ISSUE 2

£1.25

EDITED BY RICHARD ALEXANDER
PUBLISHED BY C.G.H. SERVICES

QL TECHNICAL REVIEW

CONTENTS

SUBSCRIPTIONS: £5.00 FOR 4 ISSUES, FROM C.G.H SERVICES, CWM GWEN HALL, PENCADER, DYFED, CYMRU SA39 9HA (TEL: 055-934-574)

EDITORIAL

Welcome to the second edition of QL Technical Review. I must say that I am somewhat overwhelmed by the quantity and quality of the material sent for this issue. It has meant that production could be got underway more or less on schedule. Let's hope that the printing side of things works as well.

O.K. what's in this issue ? We've got in-depth reviews of the latest version of QLiberator, and ARK's The Spy; a piece to get us started on pipes, another on using When Error and the beginning of a series of articles evaluating business software on the QL. I've let John Silk write about Textidy, as it may well be of interest to other readers and there's a brief history of PC emulation on the QL. We've also got two pieces on the Mandelbrot set for those of you wishing to investigate this fascinating area. We've also got a look a the Minerva ROM from a user's point of view.

Now what is there in the pipeline for the next issue ? Several people have mentioned that they would be writing articles covering, amongst other subjects, the new Minerva ROM, Lightning Special Edition, Text87, and we hope that Rich Mellor will find the time to continue the machine code tutorial and that Michael L. Jackson will have settled into his new home and be able to continue his look at business programs.

So far, so good. What would I like to see covered in future issues of QLTR ? Firstly a guide to using specific printers. Although both QL World/User and QUANTA have covered this area, the items have usually been very spread out, so if you can help here, get in touch. In particular setting the printer up to do screen dumps, listing progs and interfacing with the Psion progs are the subjects most in need of attention.

Another area we should be covering is educational software for the QL. If you, or your children can recommend any software we'd like to know. I suspect that quite a few QL's come into contact with small people and that they'd appreciate some interesting software.

Also if you've written any programs that you would like to share with our readers do send them in, especially if you can talk us through them. Even useful procedures etc may help somebody.

This issue should have included more about communications on the QL. However, if we are lucky we should be getting an article next time by Dave Fullerton, who has helped set-up the Bulletin Board for the Bristol User Group, showing the pitfalls to be avoided, sources of helpful information etc. Dave may also be persuaded to do a guide to using QBUG and other QBOX systems.

Inevitably we're going to have problems with the level of the writing in this magazine as people will be bringing to it different competences. All I can ask is that, at least for the first few issues, contributors take care to explain what they are talking about and not assume that everyone is as clever or experienced as they are. Similarly if people read things which go over their heads, do write in and I'll see if I can get the authors to backtrack a little. However, don't expect writers to paraphrase the QL User Guide.

Whilst we're talking about QL publications, it would quite useful if people could recommend books on their areas of speciality. Not all QL books are still in print, but you should be able to pick up some on the second-hand market or obtain them through the public library.

A point about copyright and the Public Domain Library that CGH Services runs. I'd like to be able to put all programs that are pulished in this magazine into the Public Domain. As we don't pay for the articles you send in (not even several months later, like another publication I could mention) I consider that everything in here remains the copyright of the original author. However we are keen to broaden the P.D. library, and as readers will be able to type the programs in for themselves anyway, I'd like to ask all authors to give the O.K. to making their progs in here Public Domain. Note, if anything is labelled " it should not be distributed commercially or without acknowledgement of the author or place of publication or reproduced in another magazine without the author's permission.

The deadline for the next issue is March 1st - but if you can't quite make that date, do send your item(s) in as soon as possible - it might make the difference between QLTR 3 coming out and being delayed another month or so.

Richard Alexander (Editor)

NEWS

Having a somewhat leisurely production schedule means that it is difficult for us to very up-to-date with news etc. Matters are made worse by the fact that we are not yet established as a recognised publication and so do not receive details of new programs or hardware. So if you read all the QL mags, you may well have seen most of these bits before.

MINERVA ROM

QUANTA, in their latest issue carries quite a detailed article by the authors of the MINERVA ROM. I understand that many of the bugs have been ironed-out and that it now represents an excellent value for money upgrade for the QL. However, for some of us who do not like having to pull our computers to pieces evry few minutes, it means buying another machine just to make sure that software is compatible with the new ROM. (JM, JS and MINERVA are all well represented among QL owners (or will be soon) and there are many THOR machiines as well. Ideally software should run on all of them without any problems. But it is only by actually having the machines up and running can one test them.)

3RD ALTERNATIVE MICROSHOW (STAFFORD)

Well, I actually got to this one, thanks to Hilary. Very nice to meet Rich and Jean-Yves and a few other punters, but really it was dismal. A huge hall but 2.5 miles from the centre of Stafford and the weather was wet. In short, little business was transacted that day. Let's hope the Leyland bash does better. More importantly we need a single day event in London to sell our goodies at. I understand that there will be no more ZX Microfairs. Anyone fancy organising a "Sinclair/ Amstrad" show in London. A well-heated church hall next to a tube station would do nicely. Or a community centre. We'd need space for about 30 stalls, £20 a stall would pay for itself and the advertising - especially if exhibitors could be persuaded to do some. A nice little earner I reckon. Provided you did so well in advance, you'd probably get all the relevant publications to give it a plug too.

NEW QL PRODUCTS

As and when somebody sends us the details we'll pass them onto you. Even a spec. sheet can be edited to provide some news for people.

QL WORLD

Far be it for us to start crying wolf, but it seems that QL World are in financial difficulties. Apparently, despite hiking their already excessive advertising rates, and being very quick to follow-up on their invoices, they are finding it very difficult to pay their contributors, people having to wait up to 6 months before the money is forthcoming. Indeed they have now informed people that it may be another 4 months before any money is paid. Plainly the mag is getting closer to closure, not that it is in our interest to lose it as they have been very good at reviewing our products. Incidentaly, the rates of pay (assuming it eventually turns up) for authors of reviews is greater than the royalties we have paid out to the authors of the games themselves so far. Maybe it is time for all interested parties to get together to sort out some contingency plans!

THOR RUMOURS

It seems that this may well be on its last legs as Amstrad have been checking the code used in the operating system and Alan Sugar was not amused. Also problems continued to occur due to incompatibility with some of Tony Tebby's software. However there is still a glimmer of hope, in that discussions and plans for a machine not dissimilar to the ill-fated Sandy Futura are continuing. So a hardware upgrade path for QL owners may still occur. Also now that restrictions on the export of 68000 machines to Eastern Europe (and the excellent political changes there!) have been lifted, Amstrad may see the economic sense in licencing a QL based machine, especially if it involves no expenditure for themselves.

QL EMULATORS

Leon Heller has reported in New Computer Express that there is now a Public Domain emulator for the QL that runs on the Amiga. So there's a much cheaper upgrade path, rather than buying an ST and then forking out £155 for an emulator. We'd like to add this to our library and may be able to arrange for copies to be made through our contacts. Apparently it comes on 3 disks, 1 for the QL and 2 for the Amiga. More news when we get it. (I.e. when you send it in!)

Richard

THE CHOICE IS YOURS

EVALUATING QL SOFTWARE PACKAGES

Though quite a number of QL software packages are no longer on the market, it is still rare for there to be only one program offering the features required by the user. Usually there will be a choice of such programs, whether word processors, text editors, graphics or CAD packages, assemblers, compilers, screen dumps or desktop publishers. Potential sources may include professional suppliers, Quanta or QL SUB libraries, Microdrive Exchange, CGH public domain or second-hand copies of 'golden oldies'. In fact, good availability of software may even delay your purchasing decisions if you tend to find it difficult to make up your mind when spoilt for choice. Just how do you choose between a full price package costing say £79.99 and a program produced by an amateur enthusiast sold at a nominal price of £4.00?

This latter question is of immediate relevance to me as a software reviewer since I intend preparing a series of articles examining packages ranging from 50 pence, plus a blank disk, to over £100. It is because the programs will cover such a wide price range and offer quite different advantages and disadvantages that I feel it necessary to start by preparing basic 'rules' to enable software packages to be judged objectively. I aim to evaluate programs of a financial or business nature, but the principles can be applied to any software, including games. Rather than discussing features desirable in financial programs I will look now at those which all excellent QL software should possess.

The features which the user requires in a program should be ordered into a hierarchy, with weights attached to each aspect to reflect its importance. An evaluation should be carried out for all programs which have the potential to meet the user's needs, with a mark out of ten being awarded for every feature. If the score of all the software packages was the same then the final purchasing decision could be made on the basis of lowest cost or easiest availability. It may be useful to set out the results on paper in a matrix.

DESIRABLE FEATURES

I suggest that it is worth looking at the following features when comparing QL software of whatever type. I am sure that most readers will be able to add their own ideas based on their past experiences and future needs. The order in which I present the points is not intended to to be a hierarchy of importance for all programs for all QL users. Everyone should set their own priorities. I do not intend naming specific programs in every section as readers will think of appropriate examples from their own collections.

1) MEMORY REQUIREMENTS

Some software will not work with more than 128K memory. Though there are professional or PD programs to reset memory to 128K RAM, having to do so must be quite a nuisance when you have paid £200 to £300 for a Trump Card or similar memory expansion. The 128K RAM may not be sufficient to allow the program to operate with disk drives, RAM disk or a printer buffer in memory. An arcade game in the Quanta library will not work correctly on expanded QLs even if memory is reduced to 128K and there may be programs elsewhere with this defect. Software which requires 128K maximum may not seem a problem if you have an unexpanded QL, but are you absolutely certain you will never find a bargain 512K at £30? You may have to abandon some older programs and purchase ones more flexible in their memory requirements.

Other programs will not not work without memory expansion, usually an extra 256K, i.e. 384K RAM in total. Most of these programs offer more features than could fit into an unexpanded QL and may be sold with a title such as 'special edition' or 'extra'. Hopefully the advertisements will mention the need for extra RAM. If you have memory expansion it may be worth paying a little more to get the extra facilities.

Good programs should be able to take advantage of whatever memory is available to them, offering the user say a bigger spreadsheet size on a 640K QL than on a 128K one. Some programs work correctly on expanded and unexpanded QLs but offer no improvements to users who have bought memory expansion.

2) COPY PROTECTION

Fortunately most recent QL software has relied on the protection of the Copyright Act rather than methods which may hinder the usefulness of the product. Protection methods have includes ROM dongles, lenslocks, orange manuals and use of original cartridges as a loading key. This latter protection method is particularly dubious as some users may accidentally corrupt their master cartridges, thus rendering their software unusable without further expense.

A number of utilities are available which allow copies to be made of the master cartridges. These are not illegal under the Copyright Act but must not be used for software piracy, which is dishonest and deprives QL programmers of a decent living. Try to avoid software which is copy protected, but if unavoidable make secure backups.

3) DRIVE USAGE

Some programs, mainly older ones, have been designed for microdrive-only operation, while some more recent ones work only with disk drives. In the latter category would be programs written for the Thor which also work on expanded QLs. Disk-only Thors will not be able to use some old programs which cannot be converted to disk operation, which are mainly copy protected games. Choose the software appropriate to your hardware.

To convert software to work from disk may require a string replace utility, to change all drive references from 'MDV' to 'FLP'. Or the disk interface may offer the 'FLP_USE MDV' command. A few programs may still insist on a key cartridge in mdv2_. Obviously something like a cartridge recovery program would be inappropriate to disk usage in any case.

The make of disk interface on your QL may influence choice of software. PC emulators, disk doctors and file transfer programs may rely on the interface having a Tony Tebby/QJump ROM, which offers direct sector access. Sinclair/Micro Peripherals owners can buy a replacement QJump ROM to upgrade their interfaces to current standards at a cost of around £15, but MCS and Quest owners have problems and will not be able to use certain types of software. Also the Quest interface uses an unusual FORMAT command and will not be able to do a format from within most packages. One small plus for Quest owners is that they can run CP/M-68K, reviewed in QLTR #1.

Now that hard disk drives are becoming available for the QL, owners will have to consider whether they will be able to use all software with one. Some programs can be copied from protected cartridge to a disk, but this relies on formatting the disk with a hidden security number. I assume that such programs could not be used on a hard disk.

4) SCREEN DISPLAY

QLs are capable of being used with everything from a black and white television to a high resolution colour monitor, without modifying the computer's hardware. Certain programs tend to assume that the user has a full 85 column monitor, not a 64 column TV, though I would expect this to be most likely of software aimed at businesses. Even so, a small trader wanting to enter the world of computing may not want to spend £300 on a monitor in order to be able to use certain programs. The approach of the Psion suite in allowing use in 40, 64 or 80 column mode is to be approved of. Some packages come with a configuration utility to allow window size and position to be changed permanently and others let this be done at run time.

5) PRINTER COMPATIBILITY

For some users the most important feature of a program will be the quality of the printed output it produces. Unfortunately some packages are a little inflexible in their printer requirements, requiring an Epson FX80 compatible dot matrix to be attached to serl. occasionally programs will not work with a parallel printer port such as that found on a Sandy SuperQboard, which gives the 'PAR_USE SER' command and use of a printer buffer in RAM. In addition, many printers do not emulate the FX80 which the program demands. Try to look for programs with configurable printer drivers - a few are capable of driving 24 pin printers and lasers to give top quality output. Or if a graphics or DTP package does not have a good screen dump it may be possible to save screens to disk and print them later by using one of the more flexible printing utilities now available from several suppliers.

For help with printers, send a SAE to Nick Godwin's new user group, Write On-line, 4 Harkur Crescent, Eyemouth, Berwickshire, TD14 5AP.

6) PROGRAM AVAILABILITY

Some of you may not agree, but I think the disappearance of QL software from High Street stores does have one beneficial effect - the end of impulse buying. Though it was nice to see a program on the shelf, only a very superficial impression could be obtained since very few shops would offer would be purchasers the opportunity to try programs before buying. Sometimes fairly mediocre software was impressively packaged, which could be misleading. Having to send off for software from a mail order supplier should tend to introduce more thought into the process, so that the program available from a local shop will no longer be more highly ranked than one which will take a fortnight to come in the post but might be better in all other respects.

If the supplier is based in Europe or America, with potential problems in respect of currency and other risks, then such programs would receive a lower score than those available from a UK source known to be reputable. It might be better to wait until a supplier you have dealt with before starts to import the software, thus taking the risk on your behalf.

A particular program may acquire a 'scarcity value' if it has not been available for a long time and a small supply is advertised. As availability is likely to be very short term it might be better to buy quickly while you have the chance. On the other hand, the lack of availability may have been because the product was less successful than its rivals. The original programmer may no longer be supporting the product and it may have bugs which will never be corrected. Only award points for scarcity if a program is at least equal in all other respects to those programs being supported adequately by their advertisers.

7) PROGRAM COMPATIBILITY

Many QL owners have one or more ROMs, containing toolkits plugged into their computers or on disk interfaces. Others load such software from disk at the start of a computing session. They need to be confident that new programs they want to buy will be compatible with existing toolkits. Sometimes different programs use the same identifiers for quite different purposes, for instance a BASIC program might define a procedure with the same name as a command on the Toolkit II ROM, which will generate an error message such as 'bad parameter' on QLs with TK2 fitted. The problems mainly arise with amateur programs but I have had trouble with professional software.

The editor has mentioned before the desirability of co-ordinating use of program identifiers so that there are fewer clashes with the commands contained in the major toolkits. Programs should be checked to be compatible with all the major QDOS ROM variations, at least from AH to JS, but preferably also MG(UK), Minerva and Thor XVI Argos operating system, and even the ST-QL emulator. However, few individual reviewers will have access to all of these so QLTR readers should keep the magazine informed of any successes or failures worth mentioning.

8) MULTI-TASKING ABILITY

This is a similar issue to compatibility with toolkits and ROMs, but relates to whether a program can run alongside one or more other tasks. Even the Psion programs can take advantage of multi-tasking, without any special tricks. Small utilities, such as printer spoolers or screen dumps, can be started with the EXEC command before running say Quill with EXEC_W. Or BASIC programs started by LRUN can also have other jobs working with them. But some 'badly written' software demands the QL's full attention and will not even load if any other jobs are running. The main culprits are machine code programs loaded with LBYTES rather than EXEC, and protected BASIC programs started with the FLRUN toolkit command.

Some programs can be made to multi-task by using one of the many job-swapping utilities, such as QRam or Taskmaster. Others will do this quite adequately by running them with EXEC and pressing CTRL-C to swap to other tasks. Several good programs have a screen refresh key press, so if the window of a job is overwritten by others, when you swap back to it you can redraw its screen by pressing say F4.

You may encounter certain combinations of programs that do multi-task but do not work well together. Unfortunately such problems are likely to be obscure and you may not be aware of them before purchasing software.

9) NETWORKING CAPABILITY

Some users will have two or more QLs and may be interested in whether a particular program can use the network. I will assume that all QLs to be networked have Toolkit II on ROM, so that at least one can operate as a file server for the other(s). Many programs will be able to use the network with the NFS_USE command. For example, NFS_USE FDV,N3_FDV1_,N3_FDV2_ will direct access to the FDV drives to FDV1_ and FDV2_ on network station 3, which is useful when a program insists on a five character drive reference. Other programs are more flexible and will accept any valid device, such as N3_FDV2_. But some programs, though they can be loaded from one QL to another across the network, simply ignore any further references to the other QLs on the network.

Incidentally, I have never come across any QL program that has been explicit in any sort of licence agreement about the terms of its use on a network. In theory you pay for the right to use the program on one QL at a time, so if you wanted to run it on three QLs on a network you should buy three copies of the software (unless a 'site licence' is offered, whereby you are licenced to make as many copies of the program as you need for use in your premises). Most QL software houses would tolerate you running one copy of their product on two or three QLs so long as this was not abused, but I do not suppose that there are any large QL networks so nobody will be losing out.

10) ERROR TRAPPING

We all make mistakes, but the punishment handed out by some programs can be a little excessive. Look out for good software with error trapping routines which let the user recover from slip-ups. Programs compiled with Q_Liberator at least offer a 'Retry Y/N' option for most errors if the programmer has not added specific error traps. Reviewers should try to crash programs in several ways in order to report problems which could arise for typical users.

11) USER INTERFACE

The look and feel of a program is a very subjective feature, but is worth considering. While some users may desire the use of icons and a pointer, others like text menus, yet more prefer to remember quick key presses without help prompts to slow them down. The Psion programs offer a compromise between these three variations, since they can work with prompts on or off and even with the icon-controlled Icicle. There is a vast range of styles of presentation of programs so most people should be able to find something to suit them. However, if you want everything to have the same user interface, perhaps you should buy a Macintosh.

12) QUALITY OF MANUALS / DOCUMENTATION

Whether a good manual is important or not will depend on a combination of factors, principally the ease of use of the software and the general experience of its owner. One icon controlled graphics package comes with a tiny printed pamphlet but because most operations are self explanatory it does not require anything larger. Other programs are accompanied by nicely bound 150 page manuals which you may have to refer to frequently because of lack of on-screen guidance. In general you should not expect printed instructions with 'library' or public domain programs - you might get a Quill _doc to print out on your own printer or read on screen. More expensive software should be supplied with printed instructions, though one £50 package requires purchasers to print out the manual! Other suppliers send manuals full of advertisements for their other products, printed on orange or red paper so that you cannot photocopy them (or read them?), and you have to find your own binders. One supplier charges an extra £4.00 for a 28 page photocopied manual which should have been sent free of charge in the first place.

If you do not have a printer or do not progress well with programs without adequate documentation to read then award extra marks to software which has good printed manuals.

13) PRICE OF PRODUCT

The cost of software should not be the most important factor in determining purchasing decisions, otherwise who would buy professional programs at say £89 if there is something free in the Quanta library which 'almost' does the same thing? The use you intend making of the product will influence the price you should be prepared to pay. If your financial well-being depends on it and you want the best features possible, then if it costs £130 -pay it. But if you only want to dabble with as many different types of program as possible, 'bargain' software will probably be enough to give you the sort of experience or amusement you want. I have bought programs which used to sell at £25 to £60 for £3 to £5 from classified advertisements; though I know better programs are available, I have been quite happy with my purchases.

If a program meets your needs and is also a bargain then it could be given extra points.

14) REVIEWS AND OBSERVATIONS

To gain an insight into the features possessed by programs it may be necessary to rely on reviews in magazines such as QLTR, QLAF, QL World, Quanta, QL SUB, QL Forum and Computer Shopper, or on certain QL bulletin boards, or word of mouth reports at Quanta workshops or computer shows. A program which has received good reviews from several sources should be awarded extra points, but I suspect that some very worthy programs have never been reviewed at all. Occasionally magazines are very enthusiastic about any new program if the supplier regularly advertises in them.

Sector Software publishes a useful index program for most QL User and QL World issues and there is a database of program details available from Quanta; these should help you find the magazine issue with that vital review!

Attendance at a workshop may be the only chance you get to try a program before buying it and Quanta has been very active in this field in recent months. Members set up informal demonstrations of programs they use, so that you can lean over their shoulders and see how things should be done or even ask for a go. Others take along software they are having problems with to see if anyone can sort things out - sometimes the program authors are present to help resolve difficulties. I would award high points to software seen in action and known to have enthusiastic users and support from the programmers.

15) CREATING AN EVALUATION MATRIX

Not all of the above points will be of interest to all QL users. Choose four or five main points which you consider important and assign weights to them. For each program of a certain type you are considering buying award marks out of ten for each factor, then calculate the weighted total.

Take the example of programs used for maintaining a database of knitting patterns, as supplied by "Digital Incision"(A), "Sexual Software"(B), and "Teacake and Pewterware"(C). Four main features are considered desirable: 1) must work with memory expansion; 2) must work with single Sinclair disk drive (device fdk1_); 3) must have configurable printer driver to work with Oki Microline 193; and 4) must have adequate screen display on a monochrome monitor. The results are as follows:-

		Supplier A		Supplier B		Supplier C	
	Weight %	Score from 10	Weighted score	Score from 10	Weighted score	Score from 10	Weighted score
1	20	8	1.6	2	0.4	10	2.0
2	20	10	2.0	6	1.2	3	0.6
3	30	5	1.5	10	3.0	4	1.2
4	30	8	2.4	2	0.6	10	3.0
	100		7.5		5.2		6.8

As the programs are similarly priced, you choose supplier A's product. However, if you spot it at a good discount, you might also consider supplier C's program.

I cannot pretend that I have never bought a dud. I hope I never recommend one!

Michael L Jackson

WHEN ERROR

Below is a short program from Ralf Biederman of Langwedel in Germany. It deals with the WHEN ERROR command. Please note that this program is only applicable to JS or MG ROM QL's with Toolkit II enabled. Ralf is the author of quite a few useful programs, which can be found on a disk in our Public Domain library. Cost is a mere £2.00 (plus media), but a word of warning, many of the programs are in German. (No problem for our German readers obviously.) Ralf is also the "author" of QL XLISP which is also in our Public Domain, and which we hope to have working 100% very soon.

continued overleaf...

WHEN ERRor ?

Most people use the WHEN ERRor routine on the QL only to prevent their programs from crashing, but you can do a lot more with this beauty. I always found it annoying to add EX or EW and the drive to a Job I wanted to start. In some of my other programs I overcame the problem of the drive but I still had to add the procedures EX and EW. To use this utility you must run it first, after that the error report BAD NAME is trapped by my routine. BAD NAME is generated when you type in a name which is not a Basic procedure. In my routine I get the name that caused the error, after that I look on FLP1_ and FLP2_ to see if there is a Job with this name. If I find one, I start the Job with EX. If there isn't a Job, I look for a file with the name and extension "_BAT". If there isn't one, I merge it with my routine. In a file with the extension "_BAT" there should be a Basic program without line numbers, so they can be executed directly. In a file like this you can use things such as DATA_USE or WINDOW to suit your wishes. This program is only an example, you can add things like loading new Basic procedures with LRESPR or whatever you like. I think that you have something like an MS-DOS feeling. To test the routine, type it in, run it and then type the name of a Job, which is on disk, but forget to add EX and the drive. For example type in: EDITOR (ENTER) and look what happens.

WHEN ERRor Listing Ralf Biederman

```
32000 TK2_EXT
32010 WHEN ERRor
32020 IF ERR_BN THEN
32030   adr = PEEK_L (HEX ("28010"))+HEX("168")
32040   name$ = ''
32050   REPeat loop
32060     zw = PEEK(adr)
32070     IF zw  = 10 THEN EXIT loop
32080     name$ = name$&CHR$(zw)
32090     adr = adr+1
32100   END REPeat loop
32110   IF FTEST ('flp1_'&name$)=0 THEN
32120     OPEN #3,'flp1_'&name$
32130     job = FTYP(#3)
32140     CLOSE #3
32150     IF job = 1 THEN EX 'flp1_'&name$
32160   ELSE
32170     IF FTEST('flp2_'&name$)=0 THEN
32180       OPEN #3,'flp2_'&name$
32190       job = FTYP(#3)
32200       CLOSE #3
32210       IF job = 1 THEN EX 'flp2_'&name$
32220     ELSE
32230       IF FTEST('flp1_'&name$&'_bat')=0 THEN
32240         MERGE 'flp1_'&name$&'_bat'
32250       ELSE
32260         IF FTEST('flp2_'&name$&'_bat')=0 THEN
32270           MERGE 'flp2_'&name$&'_bat'
32280         ELSE
32290           REPORT -12
32300         END IF
32310       END IF
32320     END IF
32330   END IF
32340 ELSE
32350   REPORT
32360 END IF
32370 END WHEN
32380 DEFine PROCedure sa
32390   SAVE_O command_bas
32400 END DEFine sa
```

MANDELBROTS

This is an article Carl wrote about the Mandelbrot Set, as background notes to his program. I've printed it, as it explains the maths aspect of the mandelbrot set very clearly. Carl's program is available from the CGH Services Public Domain library.

An Introduction to The Mandelbrot Set

The Mandelbrot set an example of what is known as a fractal and a pattern formed from a simple equation, is one of the most complex objects in the whole universe, it is also, perhaps, one of the few things that combines artistic beauty with mathamatical precision. You can take any point on this 'bug like' object and maganify it and you will see more 'bug like' objects, then you can repeat the process 'ad infinitum'. In this article I am going to explain how the set is produced and why I find it so enchanting.

If we imagine that the Set is the size of our Earth, the average microcomputer is able to zoom in, past the size of a country, a city, and buildings to the size of an A4 sheet of paper! Because of this you are almost guaranteed to see a part no-one has ever seen. You alone are the only one ever to see that 'seahorse' or spiral, or tail or swirl.

Before I introduce the equation, which produces the Set, I am going to show you a new type of number, a complex number. A complex number is made of two parts, a real number part, and an imaginary number part. It takes the form:

$$a + bi$$

Where a is the real and b the imaginary number. An imaginary number is simply the square root of a negative number, for example, if we want to work out the square root of -16, we know it is not 4 or -4, but what we do know is that it is equal to:

$$sqr(-16) = sqr(16) * sqr(-1)$$

So we say that the square root of -16 is equal to 4i there i equals the square root of minus one.

All complex numbers can be represented on a diagram (called an Argand diagram), which represents what is called the complex number plane, rather like a graph and the a and b (from a + bi) act as co-ordinates to this 'graph'. That is, the real number part is used for the horizontal axis and the imaginary number part for the vertical.

The equation used to generate the set is:

$$f(z) = z^2 + c$$

What that means is that you take a point on the 'graph' and call it the number c (remembering c represents a+bi). You then set z to 0,0 and iterate (repeat) it in a kind of 'feed back loop' each time taking the result form the last time and setting z to it, squaring it, then adding c (the co-ordinates of the point you are testing).To clarify that I shall replace c by a + bi and z by x + yi, the formula then looks like this:

$$(x + yi)^2 + a + bi$$

and multiplies out to make:

$$x^2 - y +2xyi + a + bi$$

and if we take the co-ordinates for the points on our graph we get:

$$X,Y=(x^2 - y + a, 2xy + b)$$

You do this for a specified number of times, or iterations. If while you are doing it X,Y goes above 2,2 then you colour the point you are testing in proportion to the number of iterations, for example yellow for points which take a long time to go above 2,2 and blue for those that go very quickly. If after you have done the specfied number of iterations the point is still below 2,2 then you colour it black, and it is said to be in the Set.

To generate the set you have to test lots of points between -2 and 2 (across) and -1.5 and 1.5 (up). You can then start to adjusting things like the width of the picture, for example by only testing points between -1 to -0.8 (real number axis) and 0.5 to -0.5 imaginary number axis.

And here, below, is Carl's instructions on using his Mandelbrot program.

A Guide to QL Mandelbrot Ver 1.1

Instructions

To produce a Mandelbrot Set select 'new fractal' from the menu.
When it says filename, you specify the microdrive (MDVn_) or disc (I am poor so I use microdrives!) and give it a file name for example:

```
mdv1_mandel
```

Do not add any thing after the filename, the program will add _pic for the picture and _dat for the parameters you are about to type in.

Xstart: This is where the program starts on the X or real number axis.

Ystart: This is where the program start on the imaginary number axis.

Res: This is how big each step is in pixels. 1 is the max resolution (1 pixel 256*256), you must specify it as a power of 2 (for example 2,4,8,16,32,64 etc). This is useful for a fast preview mode, try 4 for a preview.

Iterations: This is how many iterations the prgram will do before giving up.

Once you have finished typing in all the parameters, they are saved on a _dat file then the drawing begins, make sure you have at least 32k free (about 75 sectors) to store the picture.

It may be helpful to point out that in the display the y axis is reversed, rather like putting a mirror along the x axis. When you specify x and y, you are specifying for the top left of the screen, but as the program goes down the screen y increases, as it goes across x increases.

If you would like some pictures, send me a cartridge and an SAE and I will record three (pics) on there for you (if there is enough demand I will place pics of the set in the public domain). If you want to know about version 2, or any other aspect of programming or have some criticisms, hints or tips for this or the new version, I want to hear from you, please write, or phone(after 4pm, I'm at school), my address and telephone number is:

Mr Carl. L. Cronin
21 Greenway
Manor Park
Letchworth

Herts
SG6 3UG phone 0462-670177

Before I say bye, I would like to say a few thank-you's:

Thanks Mr Knight (who I met at the Alternative Micro Show) for compiling the program, and thanks to Mr Moran for providing the maths functions.

BYE

Further Infomation

There are many types of fractals, each of which can provide many hours of fun. The best thing you can do if you want more infomation is subscribe to Fractal Report (a bi-monthly magazine):

Fractal Report
Reeves Telecomminications Labs Ltd
West Towan House
Porthtowan
Truro
Cornwall
TR4 8AX

It's also good because the Editor uses a QL and Thor XVI!

You could also read Mandelbrot's books ('The Fractal Geometry of Nature' & 'Fractals Form, Chance and Dimension')

The best book to read is:

James Gleick CHAOS:Making a new science.

If you are rich you could buy:

The Science of Fractal Images (at about £30 is the BEST book)

New Version Soon!

I hope to have a new and much improved version of this program out soon. It should have the following features:

Directory of any device (Sorry I fogot about that because I only have microdrives)

Optional Stipples (for those of you who have TV's (I have an Amber Monitor))

Faster drawing (If I can get hold of this new algorithim)

Selection by cursors of area to enlarge

Absolutly no GOTOs or GOSUBs, the present source code is a mess (this was my first ever QL program!).

It will also be able to draw the Julia Set.

We are slowly amassing quite a collection of mandelbrots, julias and other fractal programs in our Public Domain and Shareware library. We are doing this with the co-operation of John de Rivaz and Fractal Report. The more the merrier! I'd be quite happy to print any spectacular screen dumps in future issues of QL Technical Review.
Richard.

TWO LETTERS

LETTER FROM JOHN SHAW

3 Barn Owl Way
Stoke Gifford
Bristol
BS12 8RZ

Dear Richard,

I must say that I found the first issue of QL Technical Review very interesting, particulary your article on Toolkit II. How about a regular TK2 feature explaining for the benefit of lesser mortals, some of the more useful commands with clear examples?

e.g. What is a wildcard and how can it benefit me?

Having a TK2 on my Trump Card, is there anything to be gained by calling it up in the boot before every program?

Can we have a boot listing which gives you a screen dump choice using different hotkeys etc ?

Also I have just fitted the new Minerva ROM. It seems very fast, but I don't feel competant enough to do a tech. review.

Best wishes,
John Shaw.

P.S. I have just been elected chairman of the new Bristol QL Users Club.

ED: Thank you for the compliment John, we hope that the standard of QLTR will be maintained in this and future issues, although as we're dependent on readers also being writers - it's up to the readers how good this magazine becomes.

With regard to another Toolkit II article, I'd be very happy if one of our readers could provide one that was even simpler than Mike Lloyd's one in the December QL World. (Where did the idea for that article come from I wonder ?)

As for the Minerva ROM - John Vennard has given his initial opinions in this issuer, but I suspect that the ROM will prove to be quite reliable eventually. In the meantime I'm quite happy to print other users opinions in the next issue or two. I'd personally like more reports of compatibility with existing software.

Richard

LETTER FROM JOHN SILK

Dear Richard

Congratulations on producing a new QL publication; I hope your courage and industry is rewarded.

I shall restrict myself to commenting on some queries and problems raised in QLTR1.

THOR XVI - version 6.40 ROM is virtually unusable; try 6.39, 6.41 or better, 6.42, assuming you can obtain them from Dansoft.

Quest and Microperipheral interfaces avoid them except as museum pieces. Neither are capable of "direct sector accessing", thus making it impossible to use ceratin software, e.g. LAZARUS, DiscOVER, Hardback & Finder, Media Manager, DiscED. In addition the MP interface names its disc devices "fdk" rendering some software inoperable without changing flp (or mdv) to fdk within the code.

MetaComCo's C Compiler - Because of a feature in this, the interaction of the "printf" command with Speedscreen or standard Lightning paints paper and ink the same colour. The C Compiled program runs faster but becomes unreadable to the naked eye.

Programs affected include TexTIDY and (Multi-DiscOVER). The good news is that the new rommable version of Lightning has been programmed around this feature. I believe that the Speedscreen author has also addressed this problem.

However, if using a multi-tasking MetaComCo C compiled program (i.e. where you use the EXEC command to get it up and running), you can get the best of both worlds by switching off Speedscreen or Lightning before executing the program. Once the program is up and running use CTRL + C to get into SuperBasic, switch on the speed enhancer, CTRL + C to get back to where you were and all is well.

More good news: PDQ-C compiled programs do not enjoy the printf bug.

Yours

John Silk

PDQL, Unit 1, Heaton House, Camden Street,
Birmingham B1 3BZ

TexTIDY - an overview

TexTIDY was conceived following feedback from DiscOVER users.

This modestly priced utility (£10) is dedicated to converting code-ridden text files to plain text - with a few extra bonuses.

The original problem to be addressed was that Quill _docs had to be written to file as _lis files before DiscOVERing to alien format and importing into any word-processor at the other end. If a number of Quill _docs were to be DiscOVERed the time taken became intolerable.

As TexTIDY was being built other day-to-day problems emerged.

Wordstar files DiscOVERed to the QL were only partially readable in Quill or the Editor; when sent to the printer, the final character in each word was printed in a different font, typically italic.

The reason is that Wordstar files do not incorporate a space in the saved file. Instead the last character before a keyboard generated space consists of 7 bits to identify the (ASCII) character and the eighth bit (not normally used) to denote a trailing space. Thus Wordstar tidying was added.

Once in use it became clear that at last QL users could enjoy a workable and cheap programming environment - read on for more!

TexTIDY in Operation

1. Execute TexTIDY (written in C, compiled, multi-tasking).

1.1 CTRL + C to navigate between SuperBASIC and TexTIDY.

2. So to the Main Menu where (as in the sub-menus) on-screen prompts are available at all times.

   The Menu Window displays six Options: Directory, Convert Files, Erase Files, Alter Convert Details (the Cursor default option), Swap Convert Details and Format Medium.

2.1 TexTIDY requires a Source device and a Target (or destination) device. Thus at the foot of the Options window appears a Status line displaying the current default status:

```
Source:  Mode AUTO  Device flp1_
Target:  Mode AUTO  Device flp2_
```

   Below the window sits the command line:

```
ESC   Cursor   Space   Initial letter   Enter  Change next  Change Selection
```

   ESC offers the Y/N prompt (ENTER for Y saves a key press); you can scroll through the Menu with UP/DOWN arrow;
   Space for the next Option; or simply press the initial letter of the option you require.
   ENTER when happy.

3. Alter Convert Details

3.1 Using this Option enables you to alter either device to be used and the file type for either set of files e.g. convert QL Quill to DOS Quill, or Wordstar to text.

4. Swap Convert Details

4.1 Selecting this Option swaps Source and Target device details.

5. Directory

5.1 This Option allows you to examine the files currently resident on either device. The device size and free space is displayed. Each file is identified with date, size and type (e.g. DATA, program, QL Quill etc.). Different Directory levels, supported throughout TexTIDY, are available in this Option. There is also a View (contents) File Option.

6. Erase Files

6.1 You can view file contents, erase by marker or select whole disc (or tape) for erasure.

7. Format Medium

7.1 Either medium can be formatted from within the program with (or without) a selected (by you) name.

8. Convert Files

8.1 At last; TexTIDY doing its job.

Select your files for treatment (view first if you wish) press ENTER and before you have poured out your coffee the task is completed.

Why should you use TexTIDY

Cleaning up files ported over from non-QL formats (or prior to porting to alien machines) was the original motivation.

An interesting by-product quickly emerged. If you write (or import for editing) a SuperBASIC program in Quill - thus using familiar editing routines - you can now TexTIDY the saved Quill _doc such that it is ready for compilation or running in SuperBASIC. You can use the same technique for Archive procedure files. One point to be observed; no "long" lines greater than 127 columns - TexTIDY does not approve (nor do most serious programmers).

John Silk, PDQL

TexTIDY is available on disc or cartridge from PDQL, Unit One, Camden Street, Birmingham B1 3BZ (021 200 2313) at £10.

Ed: I've decided to give John space to promote his product as I feel that programmers and publishers should be given the opportunity to put such information in their own words. However there is a thin line between such articles and advertising and publishers are warned that in future we may be stricter on such articles.

Our advertising rates are very cheap, only £20 for a page, £10 for half a page etc, so there's no excuse for not advertising. Also we'd prefer to have our hands on products before they are advertised - or at least advertising should make it clear on the current status of the product (still being written, final de-bugging, on sale now (but there's still a few bugs or, finally, that's as many bugs as we can fix!)

Please note that we are happy to have any company's products advertised or reviewed in this magazine and are more than happy to print both adverse comments and praise where these can be justified. Please note that all opinions expressed are those of the contributors, not the editor.

Richard Alexander (for it is he.)

QL Forum,
26A Ferrers Ave,
West Drayton,
Middlesex
4th September 89

Dear Richard,

Because the accompanying article on pipes and filters was written when I was still doing my A-levels (about two years ago), it was not submitted for review.

Instead it was sent post haste to Richard Turner of QL Sub who gratefully accepted it. Now I feel that the article must have a decent chance of being printed.

It could only have been written with the help of Tony Tebby who provided up to date filters. He also looked at a copy of the article and, as he raised no objections, it must be reasonably sound.

I should have done this earlier. I intend to give filters as much publicity as possible. You can publish it first but then could I get it printed in Quanta (giving you a bit of free publicity "You saw this first in QLTR#2 etc).

Thankyou for printing the Qptr notice. I will be working on Qptr in the future and have submitted an improvement to its sprite editor 'edspr_bas' to Tony Tebby. Currently the sprite editor saves the definition as assembly language source. My alteration means it also produces a little SuperBasic function to define the sprite using the Qptr SuperBasic toolkit routines.

Ian

Programme Evolution - a Pipe Dream Too Far ?

Not too long ago a programme was a list of instructions I "typed" into my hand held computer, a Hewlett-Packard 33E. Every time I wanted to use a programme, I would type it in. But the QL with over a thousand times as much memory changed all that. Now there are different kinds of programmes & they don't have to be typed in just to use them.

So life has become a bit more sophisticated. So what? Microdrives are handy, discs are fast and hard discs even faster. But is that "it"?

There are three popular types of prog (a) Basic (b) resident procedures and (c) executable files eg Quill. And there are more. But are they useful to the user?

Resident procedures are permanent progs, staying in the QLs' memory until power down. They can be in the form of additional commands for SuperBasic, device drivers, "hotkey" programmes, almost anything. Device drivers are just progs that add devices (or replace existing ones) such as ram discs and spelling checkers. Hotkey progs are a way to have your cake and eat most of it. You can load your "hotkey" jobs into memory all in one go and whenever the programme is needed, it "loads" and runs immediately as it is already in memory. Hotkeys use a technique called "code-sharing" (aka "re-entrancy"). It means that you can have one copy of the programme instructions in memory shared between tasks to cut down on memory consumption. For example in the Qram utilities package, only one copy of the instructions needed for the Qram task is ever kept - you can have as many Qram tasks as you want (limited by a maximum of 117 as well as by memory). But there are drawbacks. At the moment there isn't any info (that I know of) that explains hotkeys so that programmes can make even better use of them. But here is a possible example on the better use of hotkeys:

Imagine a desk top publisher that relies heavily on menus, all to act on a central core of graphics information the "page" and some character sets. If you had a series of "hotkey" jobs working together on a single page, the system would be extremely flexible - the user could have two copies of the character set editor task working on separate character sets while the text entry part is busy reading a text file. Things get a little bit more complicated when you have a couple of pages being edited. Then you could have several programmes working on a page and have several programmes working on another page. And although there would be two pages in the QLs' memory, you would only have one copy of each programme. In a way I'm glad that that kind of thing isn't going on at the moment as it tends to slow things down. So a faster processor would be desirable (as well as completely unaffordable). But the main point is, the hotkey process should be accessible by programmers through other means than the user pressing ALT and a key. It would be useful if the limit of "one hotkey, one programme" was broken.

A far more mundane example would be to enable the short programmes I have stored on ALTKEYs to be (efficiently) used as hotkeys from a single hotkey task - all this would need is the ability for a hotkey task to be allocated more than one ALT key and to be told which key was pressed to start the job. The hotkey system two is a great improvement as it allows altkeys to select a task, load a task or select SuperBasic and enter a command.

Disc drives are fast enough at the moment but with all these memory-resident programmes, why not cut out the middle man and have them built into the computer, just like the QLs' rom? Then we would have problems with flexibility - erasing a block of eproms every time you want to change your set of built in programmes isn't convenient and is expensive. Enter Static Ram. Static in speed? No! Not only is it faster than the normal memory used in the QL but it keeps its contents when the power is off and is very easy to alter. So how about taking some static ram and using it like a rom board? A lot of people have 640k QLs & it would be possible to put an extra 256k of static ram on top of that. It could be used as either memory or a pseudo-rom or even a "permanent" ram disc in any combination. Static ram for the QL is not new - Sandy were going to do something involving static ram but presumably the project was killed off by profit margins.

Ok, thats enough science fiction for the moment - hardware engineers can stop laughing now. There is something very real that has been around since 1983 and is built into EVERY QL. They're called "pipes" and are fully accessible from assembly language. Pipes are memory based queues that have length (ie capacity in characters) and two channels - one for reading and one for writing data. They're used for communications between tasks and are quite fast. Special jobs called filters take info from input channel(s) and send it to output channel(s) - very quickly, if they're written in assembly language - and the speed difference is very noticeable if they're not. Despite all that, use of filters seems rare.

So what's hindering the use of filters? Programming? Ease of use?. We'll take programming first. Unfortunately, the high-level side of pipe handling isn't particularly good. SuperBasic can open a channel to an output pipe but needs additional commands to open an input pipe. Turbo charged progs can handle pipes and can filter data but are not standard Qdos filters - they need the support routine EXECUTE (which sometimes leaves channels open and isn't the same as Toolkit Twos' EX). Lattice C can use pipes but I can't use Lattice C. Computer One Pascal just ignores any pipes. If anyone has had any better luck with other languages then Let Me Know. The Qram utilities package is invaluable for de-bugging, as it will show who owns which pipes and which jobs are present. Here are some examples in SuperBasic that use Qjump's filters to do the processing.

For the examples below, a simplified definition of the EX command would be :

```
EX job_file_name, Input_file, Output_file; Parameter$
```

The input files and output files can be SuperBasic channel numbers or Qdos device names. Please only consider that as a very general definition. EX can do a lot more than start a task, give it two channels and a parameter string. The "simple" filters I know of are as follows :

Prog LC converts an input channel's data to lower case and sends it to the output channel. It may be used with :

```
EX LC,ram1_Boot,ram1_Output
```

Prog LNO adds line numbers to its input data but cannot handle long lines. It may be used with :

```
EX LNO,ram1_Boot,ram1_Output
```

Prog MORE shows a file a screenful at a time, waiting for a key press every time a screen is filled. In some cases, one job using another jobs windows can cause trouble. It's best to avoid it by using VIEW whenever possible. It's used with :

```
EX MORE,ram1_This_bas TO UC,#1
```

Prog PAGE paginates an input file with headings but cannot handle long lines. It may be used with :

```
EX PAGE,ram1_Boot,ram1_Output;"Boot on "&DATE$
```

Prog UC converts an input file to upper case and it doesn't matter what size the line lengths are. Its used with:

```
EX UC,ram1_Boot,ram1_Output
```

The next block all concatenate a number of input files and send them to a single output channel, with slight differences in the way they get their input data.
The prog CCL cannot handle long lines and is called by :

```
EX CCL,ram1_Scr_do,ram1_Beans,ram1_Output
```

The prog CCT can handle long lines and is called by

```
EX CCT,ram1_Scr_do,ram1_Beans,ram1_Output
```

The prog CCTF can handle long lines but is "special" in that it doesn't want a list of input files in the EX statement - the input file tells the job where it can find a list of filenames [it appears that the list must be on a file and not in a channel] (maybe the "F" in CCTF stands for a Folder of files). Used by

```
EX CCTF,ram1_File_list,ram1_Output
```

The prog CPY will copy an input file to many output files. It is used by:

```
EX CPY,ram1_Boot,ram1_Output,ram1_Output2
```

Filters can be set up so the output of one prog goes to the input of another prog and so on. These are called "production lines" because, like a factory, each stage adds something to the product. The prog CPY is useful as the last job in a production line - you can have the results of the production line sent to the screen to keep an eye on things, to the printer for a record in case something goes wrong and to a disc file for future reference. Now isn't that better than faffing around with COPYing to the screen and SPooLing to the printer AFTER everything has finished?

One example of a production line is adding line numbers and converting to upper case the file "ram1_This_bas" and sending it to the file "ram1_Output". Use:

```
EX LNO,ram1_This_bas TO UC,ram1_Output
```

A production line could be used as a temporary extension procedure/function to SuperBasic. Heres an example that does the equivalent of an UPPER$ converion to upper case.

```
OPEN_NEW #4;pipe_300
CONNECT #4 TO #3
EX UC,#3,#4
REPeat Convert
     INPUT #0;"Text (Quit exits)"!Text$
     PRINT #4;Text$
     INPUT #3;Product$
     IF Product$=="quit" THEN EXIT Convert
     PRINT "Converted"!Text$!"to"!Product$
END REPeat Convert
CLOSE #3;#4
```

Now I am no expert on these things - I'm just finding my way through pipes - but the above piece of code works.

But is there a use for all this? Do you really want to add line numbers to a file, paginate it, convert it to upper case and then output the results to a multitude of devices and files (would it be interesting to see it on a network for sending electronic junk mail)? Well, maybe not very often - at the moment filters seem to be written by programmers, for programmers.

There is a definite problem with ease of use. It would greatly improve ease of use if production lines could be communicated with via device names instead of channel identifiers - a kind of temporary device driver. This facility for temporary programmes is provided by the elusive QL entity known as the "thing". This allows programmes /data /anything to be shared by other programmes and there are facilities for "things" to be shared by programmes or used exclusively. I know they have been used a great deal in Qram II [Commercial break: If Qjump are ever short of beta testers, I would be happy to help out. I haven't got hold of Qram II yet]. So they may eventually become easier to use.

Looking to the future, the use of multi tasking will become wide spread with the acceptance of IBMs'/MicroSofts' new operating system, 'OS/2'. Info on OS/2 can be found in the July issue of '.EXE', books are being published on it already. One interesting difference to Qdos that I know of is that it has "threads, processes and sessions". A thread is necessary to get some instructions executed, a process is allocated machine resources for the use of its threads and a "session" has to be present for a thread to be able to interact with the user ie console or mouse i/o. I am sticking to Qdos and, if financially possible, will upgrade via the CST Thor XVI. Ignoring the (hefty) price tag, the major obstacle between me and the Thor XVI is that the operating system, Argos is not a Qjump o.s and so I am very cautious about it. I know Qram & co were developed as the o.s for the Futura but I think it would be better to have a Qram based o.s on the Thor. This would lead to the development of products that would run on expanded QLs and the Thor XVIs - this would entice more users across to the Thor.

4th September 1989:

This article was written two years ago & now Jochen Merz sells a thing manager and an eprom manager and an eprom board. What I intend to do is to fit the eprom board on my QL and put 3 * 64Kb eproms on it. On the eproms will go Qpac-1, Qram, Qtyp, Qmon and some other bits and pieces.

Ian Bruntlett

Q-Liberator (v3.22)
by Liberation Software

There can't be many QL users who could do without SuperBASIC. This dialect of BASIC contains many advanced features, making writing programs easier and more satisfying than with earlier (and many later) versions of the language. However, SuperBASIC does have one major drawback, which even its most ardent admirer would have to admit - it is slow. Although some programs can be speeded-up by rewriting in more efficient code, there remain many potentially very useful SuperBASIC programs which suffer due to the lack of speed. But this need no longer be a problem, since for several years, a number of "compilers" have been available which convert slow SuperBASIC into much faster machine code.

At the time of writing, three SuperBASIC compilers are available for the QL. Two of these, Supercharge Special Edition and Turbo, are produced by Digital Precision. The current edition of Supercharge, which was the first of the compilers to appear, removes two of the restrictions of the original, namely the Lenslok protection device and the 64K codesize limit. With Turbo, Digital Precision have taken the concept of Supercharge one stage further, making compiled programs run even faster. The price for Supercharge seems reasonable at £30, whereas Turbo weighs in at a rather steep £100, though it does include a comprehensive Toolkit, featuring many new commands.

The alternative to Digital Precision's offerings, and the subject of this review, is Q_Liberator from Liberation Software. At the time of writing, development of Q_Liberator has reached version 3.2, with an earlier release now available as the "Budget" version. The full version retails at £60, the budget version costs £30, and purchasers of the latter can later upgrade to v3.2 for £30. This upgrade path is very "consumer-friendly" - other software publishers please take note!

Much has been written on the comparison of Turbo and Q_Liberator, most of it concentrating on the internal workings of these two programs and the exact definition of a "compiler". To the end-user however, this discussion is largely irrelevant - there are just two main questions which need answered:-

1/ Will it work with my programs? i.e. How compatible with SuperBASIC is it?

2/ How fast will my compiled programs run?

Since this is not a comparative review, I shall merely state the general consensus, which is, get Q_Liberator for maximum compatibility, or go for Turbo if you need maximum speed-up.

Using Q_Liberator

At the start of a programming session, the QL should be "booted up" with a working copy of Q_Liberator, to load the Q_LIB extensions and the run-time system. SuperBASIC programs can then be loaded and edited as usual, until they are ready for compilation - a process which takes place in two distinct phases.

Phase one converts the currently-loaded SuperBASIC program into a "workfile" which is no longer recognisable as BASIC (by COPYing to scr, for example) because it is saved in a tokenised format for compilation by phase two. The first phase is activated using the new command LIBERATE "filename", which creates the workfile "filename_wrk".

Whereas the first phase of Q_Liberator exists as an extension to SuperBASIC, the second phase (the compiler proper) works as an executable program, i.e. it can be started by using an EXEC command such as "EXEC flp1_QLIB_obj". It is easier however, to use the LIBERATE command (without any parameters), which automatically executes the program from the default drive. Alternatively, assuming enough memory space is available, Q_Liberator (v3.2 only) can be set up as a resident procedure, and initiated by typing "QLIB".

When the budget compiler has loaded, it presents a command line where information about the workfile to be compiled, plus any other options, are entered.

For example, the command "MDV1_PROG -OBJ MDV2_SUPERPROG" would create a compiled program (also known as the "object" program) called "SUPERPROG" on mdv2_, from the workfile "PROG" on mdv1_.

As the compiler is working, the line number currently being compiled is displayed above the command line, and any warnings or errors found in the program are notified in an error-listing window, which also displays statistics on the object program once compilation is complete. The error listing can be diverted to another device if desired, by using the "-LIST" option (e.g. "-LIST ser1").

In place of the simple command window, release 3.2 of Q_Liberator uses a menu-driven "front end", which is compatible with (but does not require) the QPTR interface. Mouseless users can employ the arrow keys to move the cursor around boxes containing menu items, and select them by pressinng the space-bar.

When the "source file" box is selected, the name of the file to be compiled should be entered, and that filename plus the "_obj" extension will automatically appear in the "object file" box below. If a different object filename (i.e. the name destined to be given to the compiled program) is desired, then the "object file" box should be selected, and the name edited as appropriate.

All the possible compiler options are available from the front end, and can be edited or toggled on and off as required. When ready, compilation is started by selecting the "compile" box.

Programs compiled with Q_Liberator can be started using the EXEC (or EXEC_W) command, and behave like normal executable programs, except that they require the run-time system to be present. This is normally installed in the resident procedure area when Q_Liberator boots up, but it is sometimes undesirable to have the run-times separate. The "-RUN" option gets around this by linking the run-times to the object file at compile-time, producing a single, self-contained file.

Q_Liberator object files also include a line number table, which is required in programs using computed GOTOs, RESTOREs, etc [e.g. "RESTORE (1000 + n * 10)"]. When compiling programs which do not use these constructs, the option "-NOLINE" can be used to inhibit the generation of the table, thus giving shorter object files.

Another way of fine-tuning the production of object files is by using "directives". These are distinct from "options" (such as "-RUN") in that they are entered as REMarks in the source program itself. There are several directives concerned with altering the size of the data area used by the compiled program. For example, heap space can be altered from the default size by inserting a line such as "10 REMark $$heap=5000" (for heap space of 5000 bytes). This may be necessary when very large strings are being used.

Using recursive routines can often require the stack space to be increased, and this is done in a similar way to the above, with the "$$stak" directive. If the values chosen in the data area directives prove to be inadequate in the compiled program, they can be changed retrospectively using the "patch" utility which is supplied.

Other directives allow such things as assembler extensions to be linked to the compiled program, and constants to be optimised to produce faster or more concise code.

Compatibility

It is the stated aim of Liberation Software that Q_Liberator be as fully compatible with SuperBASIC as possible, and it certainly achieves more in this respect than its competitors. However, there are commands, such as LOAD, LIST, EDIT, etc., which do not make sense in a compiled program, so any attempt to compile a source file which contains these commands will not succeed.

However, if a program works correctly under the SuperBASIC interpreter, Q_Liberator will normally be able to compile it. That includes programs which make use of computed GOTO, GOSUB and RESTORE commands, and this is one of Q_Liberator's greatest strengths. Although these constructs are frowned upon by programming experts, there will be a great many people who possess programs which make use of them, and the facility to compile such programs without involving a major rewrite can be very useful indeed.

Even so, there are a few rules which must be followed to ensure that programs will compile correctly. For example, Q_Liberator quite reasonably expects every "DEFine PROCedure" statement to have a corresponding "END DEFine" later in the program, but that "END DEFine" is expected to be the next one encountered (except for conditional END DEFines, which are automatically replaced with the more correct RETurn). This restriction unfortunately precludes the use of nested procedures. While purists might think this is a good thing, the fact is that examples of this construct appear in QL World every month, so its use appears to be accepted as correct programming practise. Anyone care to comment?

The other rules state that every REPeat should have an END REPeat later on in the listing, and that a REPeat loop started within a procedure should also end within that procedure. I don't think anyone would have any quarrel with that!

All in all, Q_Liberator provides a very great degree of compatibility with SuperBASIC, even though there will always be the occasional awkward program which fails to compile at the first attempt.

Speedup

What degree of speedup can be expected when using Q_liberator? This is one of the most important questions to be addressed, but unfortunately there is no simple answer.

The standard way to compare BASICs is with the "PCW Benchmarks", an arbitrary set of small programs for which timings are obtained. Although these benchmarks can give a feeling for the relative speeds of different BASICs, they don't really give much of an idea what the differences in a full-size finished program will be.

For that reason, I have chosen (albeit arbitrarily) a few real-life examples to illustrate what degree of improvement in operating speed Q_Liberator can achieve.

1/ Initialisation routine - this segment of program dimensions an array, then reads 2000 numbers into it.

```
10 PRINT DATE$
20 DIM a(2000)
30 RESTORE 1000
40 FOR I=1 TO 2000
50  READ a(I)
60 END FOR I
70 PRINT DATE$
1000 DATA 6,3,2,1,...etc
```

SuperBASIC : 25 seconds

Compiled   : 3 seconds

Speedup = 8.3 X

Note the dramatic increase in speed.

2/ Graphics demo - this is a QL version of the "Twisted Band" program, by Alastair Watt, as published in Popular Computing Weekly, March 20, 1987.

```
10 REMark BEST IN TV MODE (TKII USERS TYPE "WTV")
20 PAPER 0:INK 2:CLS:CLS#0
30 a$=DATE$
40 SCALE 175,0,0
50 FOR x=0 TO 1800:POINT 110+(25+45*COS(x/(PI*13)))*COS(x/(PI*12.5)),75+
(45+25*SIN(x/(PI*12)))*SIN(x/(PI*12.5))
60 PRINT a$\DATE$
```

SuperBASIC : 101 seconds

Compiled : 73 seconds

Speedup = 1.4 X

There is only a small increase in speed with this program, the limiting factor being the reliance on QDOS routines (i.e. trig functions and screen manipulation), which are unaffected by compilation. To speed these up, you need a product such as Speedscreen, Lightning or Minerva.

3/ Adventure Playtime - i.e. the adventure program from the Microdrive Exchange. The command "EXAMINE PIZZA THEN LOOK" is used to measure the increase in speed.

SuperBASIC : 6.5 seconds

Compiled : 3 seconds

Speedup = 2.2 X

In this case the speedup is useful, though modest, again due to the slowness of the QDOS routines for printing to the screen in MODE 8.

4/ Ye Classical-Type Adventure - another adventure program, this time one which has been converted to SuperBASIC from a Quill Adventure Writer datafile (using the P.D. "Quill-to-BASIC" utility).

The yardstick in this program is the time taken to execute the "INVENTORY" command.

SuperBASIC : 19 seconds

Compiled : 1 second

Speedup = 19 X

The phenomenal speed increase observed here is due to the fact that the program makes use of many successive IF...THEN...ELSE...END IF structures, and relatively few QDOS routines. Without compilation, this program would be unusable.

These few examples illustrate the general rule of thumb that the degree of speedup seen on compiling a program will be in inverse proportion to its reliance on QDOS routines.

Fast compiling and compact code

As a typical demonstration of the speed of the compiler itself (v3.2), the program "Adventure Playtime" was compiled (to ramdisk) in 4 minutes and 13 seconds. Of course, compiling to floppies will be rather slower, and microdrives can take a lot longer. The budget compiler works a good deal slower in this respect than version 3.

The size of the object code is dependent on a number of factors, such as whether machine code extensions and the run-time system are attached to the code, but generally, the object code will be more concise than the original SuperBASIC. For example, compiling "Adventure Playtime" reduced it from 88K of BASIC to an object file 81K long (including run-times, and with the "LINES" option on).

If the aim is to keep the object file as short as possible (to allow it to fit onto a single microdrive for example), the following tips should help...

1/ Do not link run-times and extensions to the object code - that should save at least 11K.

2/ Turn off the "LINES" option if possible - this can save up to 50% on long programs.

3/ Use the "$$i" directive - it treats floating-point numbers as integers wherever possible, giving greatest savings where commands such as "LINE 50,50 TO 60,60 TO 75,35 ...etc" are used, since there is a saving of 6 bytes per floating-point number converted.

Extras

So far we have considered only the compilation of normal SuperBASIC code, but Q_Liberator also includes facilities to create more robust and more powerful programs.

The first step towards this is the file of SuperBASIC extensions provided with Q_Liberator. These include error-trapping, job control and communication commands, and commands to control the cursor.

However, version 3 goes much further, enabling the compilation of programs using "external" procedures and functions, and "global" variables. A few uses for these facilities are...

1/ A commonly-used procedure can be compiled as an external, and the object code linked as a resident procedure (using LRESPR for example). The procedure can then be called in the same way as a normal extension to SuperBASIC.

2/ A set of procedures can be compiled as an "overlay", which can be "loaded" as a sort of temporary resident procedure, and "unloaded" when no longer needed, freeing the memory space previously occupied.

3/ If a program is too big to fit on a single microdrive cartridge when compiled, it can be split up into several smaller sections which are compiled separately, and linked together as overlays or resident procedures from a number of microdrives at run-time.

Which version?

Owners of unexpanded QLs would be advised to get the budget version, because version 3 uses more memory, and would restrict somewhat the size of programs that could be compiled.

Although modest improvements in compactness of compiled code and speed of compiled programs are claimed for version 3, these should not be deciding factors in choosing between versions.

The major improvements of version 3 over the budget version are...

1/ Improved ease of use

2/ Faster compilation

3/ Advanced features

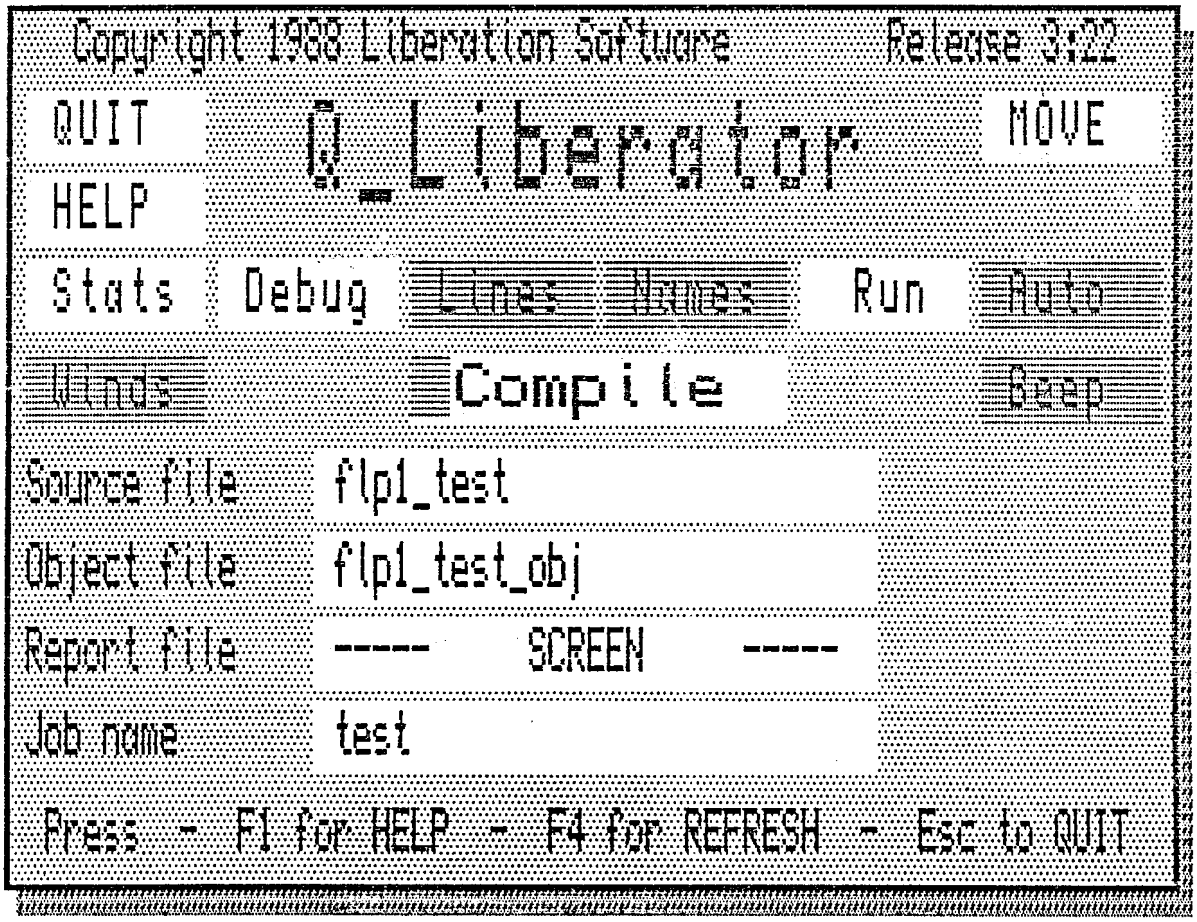
If these features appeal, go for the full version straight away, but if you are not sure whether they are important, try budget Q_LIB to start with - you can always upgrade later on without losing financially.

Conclusion

I have no hesitation in recommending Q_Liberator to anyone who programs in SuperBASIC. Offering, as it does, the ability to convert, with the minimum of fuss, virtually any SuperBASIC program into faster-loading, speedy machine code, Q_Liberator really lives up to its name.

Alan Pemberton

6/10/89



Copyright 1988 Liberation Software Release 3:22

Source file flp1_test
Object file flp1_test_obj
Report file ----- SCREEN -----
Job name test

Press - F1 for HELP - F4 for REFRESH - Esc to QUIT

Budget version: £30 (£26.95 from C.G.H. Services)
Full version (v3.22): £60 (53.95 from C.G.H. Services)
Liberation Software, 43 Clifton Road, Kingston-upon-Thames, Surrey KT2 6PJ
(01-546-7795)

PC Emulators for the QL

During mid 1988 rumours began circulating in QL circles, e.g. on bulletin boards, that some programmers were working on software that would allow IBM PC programs to be run on the completely non-compatible Sinclair QL. Though this had been achieved for 8 bit CP/M software, to do the same for 16 bit MS-DOS programs was thought to be impossible. In addition Sandy/Power Computing had been working on an expansion card to allow QDOS software to run on IBM clones and it seemed as if there may have been some confusion between this and the hoped for PC emulator for the QL.

WAR OF THE WORDS

In the December 1988 issue of QL World there was a small advertisement from Ant Computing for its IBM PC Emulator, promising some wonderful features, and a full page announcement from Digital Precision that it too had such a product on the way. The following month the first advertisement for DP's "The Solution" appeared, promising small miracles. The claims included: "The Solution runs as fast as it is possible for an MS-DOS PC emulator to run. (Does this include the Archimedes PC Emulator? - MLJ) Amazingly, it is faster than the ST one! (The ST usually is said to be 2-3 times faster than the QL - MLJ) The speed is more than adequate for everything besides interactive games." It was reported elsewhere by others that EGA graphics emulation was possible with The Solution on a QL with Trump Card.

By February 1989, Ant Computing was advertising Version Two of its emulator, claiming enhancements to the product. Version One had attracted much criticism - mainly on the grounds that it didn't work.

Meanwhile, DP had been revising its claims about The Solution's speed. For instance - "Amazingly, run on comparable hardware it is faster than the ST one!" The implication of this is that Thor XVI and ST/QL emulator users would find The Solution to be faster than PC Ditto, but not QL owners. In March, the advertising asserted that The Solution would normally run at 10-20% of the speed of a PC. Hints about enhanced graphics were dropped. More recently, the comparisons with PC Ditto have been omitted from the adverts - QUANTA found the ST program to be faster by a small margin.

Schon PCP took over the marketing of the Ant Computing emulator, but had dropped it by April in favour of Germany's The Transformer. This is said to be a hardware based emulator, offering twice the speed of The Solution. Months later it is still unavailable, but Version Two of Ant Computing's emulator has finally arrived and is supposed to work!

The war of words got out of hand at times, but general opinion in the summer of 1989 was that DP had the only PC emulator proven to work adequately. At this time I had an urgent need to aquaint myself with MS-DOS and decided I would rely on The Solution to arrange the introduction.

PC COMPATIBILITY

I will leave a technical review of The Solution to others better qualified to pass judgement. I think it is a brilliant piece of software, but not "the most important QL program ever." Its speed is adequate, so long as you are a patient sort of person. Think back to the days of the ZX81 when it took 8 minutes to load a 16K program, the screen went blank a lot, and programs on tape could cost £9.00, even if they were useless. Then the speed of The Solution doesn't seem so bad. The ability to try hundreds of Public Domain programs, available from around £2.00 a disk, tends to make up for its limitations.

It should be noted that when in CGA mode, The Solution does not use the full QL screen, only about 2/3 of it. The advertising doesn't make this clear. I suspect The Solution will be better for mainly text programs rather than those which use much graphics output. It uses the standard QL founts, so programs do not look quite the same as they might on a PC. Some seem to look better on my QL's Philips green monitor than on an IBM PS/2's black and white screen.

Digital Precision provide a list of programs which they have got to run with their emulator, or which definitely won't run (a much smaller list). I propose that a compatibility list should be maintained by QL Technical Review, to be updated by its readers. Details of the emulator's settings e.g. "timer interrupts enabled", and memory required, e.g. "needs Trump Card", might help other users. A similar list should be produced for Ant Computing's V2 emulator (it bombs - ed) and The Transformer, if and when they become generally available. I will start the ball rolling with a short list of programs I have tried using The Solution and MS-DOS 3.30.

Michael L. Jackson

COMPATIBILITY LIST VERSION 1

Program Name	Comments
The Golden Wombat of Destiny (Text adventure)	TIE
XD extended DOS	
Hard Disk Menu III v1.35	TIE
Laser Printer emulator (L.Mays)	TIE
PC Write (word processor)	
RPED full screen editor	
Autoroute+ Gazetteer Editor demo	
Various basic progs	TIE (some)
Computer Performance Integrator - cache memory/ print spooler	
PR printer driver	
IBM PC linker	
S & S Documentation Displayer	

(Note: TIE = timer interrupt enabled (slows emulation)

QLAF and MS-DOS

We shall be covering PC adventures in the new "5th Column" in QL Adventurers' Forum. Also C.G.H. Services can provide a comprehensive Public Domain service for PC adventures. We now have nearly 50 of these plus a wide range of other PC software and numerous demo progs of commercial software. Time being precious we haven't been able to test much of this on the QL. Ironically, for example, it is far better to use a CP/M emulator, such as Success, to run text adventures than an MS-DOS emulator. Unfortunately commercial considerations have "forced" companies such as Infocom to stop producing CP/M versions of their adventures. I suggest we keep the more technical material in this magazine, and the adventuring hints etc in QLAF.

Finally, after the above article was received (and not connected to its receipt) Digital Precision have started to advertise a new program called "PC Conqueror", costing £89.95 (or £139.95 for the version with MS-DOS and Windows). The advert claims that this is much faster (80%) than The Solution, allows greater memory to used by MS-DOS, greater compatibility etc. We shall wait until this becomes generally available, before making any comment. The cost for The Solution purchasers to upgrade is £50 until 31.10.1989, thereafter £60.00. This must be pretty sickening for recent purchasers of The Solution - in the September 1989 QL World the ordinary version of The Solution cost £79.95, whereas in October the price has dropped to £39.95, with no warning that prices were coming down or that a much enhanced product was about to be launched. The upside of this though is that people will now be able to get The Solution cheaper. It's a ill wind.

Richard

Quest Products - A Few Updates

SIMON N. GOODWIN - WHERE ARE YOU?

On page 13 of QLTR #1, I mentioned a bug in the driver software for the Quest floppy disk interface. I have investigated this further and would like to revise my first impressions. The problem is not a 32K file size limit after all: the Quest version 1.05 software can handle large machine code files, e.g. Quill et al, SuperBASIC programs, or Archive databases.

Difficulties seem to arise only with programs compiled with Simon N. Goodwin's 'Supercharge' or 'Turbocharge', sold by Digital Precision. I have tested such programs from the Quanta library and software supplied by PDQL and DP. In each case the result has been the same. The program can be copied to a Quest 5.25" disk but when an attempt is made to run it with EXEC, EXEC_W, EX or EW the program is dumped to the screen in #0. Example programs this happens with include 'Super Backgammon', 'Image Enhancer' and 'Better Basic'. The problem does not seem to apply to compiled programs in general as those produced with Q_Liberator seem to copy and load quite correctly, for instance Zitasoft's 'Sidewinder' or CGH Services' 'Starplod'.

There is no point in trying to attribute blame for this matter. The Quest drives were taken off the market before Supercharge and Turbocharge were first released, so neither Quest nor Digital Precision could be expected to be aware of the incompatibilities between their products. However there is a large number of programs compiled with DP's compilers, including some from CGH Services, and it would be unfortunate if Quest owners were unable to use them. Perhaps DP or Simon could explain the reason for this incompatibility and how it can be overcome (which might require some modification to the compilers or the Quest device driver). An answer in the pages of QLTR would be ideal, but SQW would be a suitable second choice. Or if an answer is supplied directly to me I will pass it on via this magazine.

CP/M-68K - SOME FURTHER THOUGHTS

Can any reader do 8" to 5.25" or 3.5" disk conversions? Some CP/M-68K computers used 8" drives and someone might be able to supply some software which I would like to try to run on the QL.

I wonder if Quest cheated when they produced their version of CP/M-68K?. In theory it 'turns off' QDOS, but in practice a few QDOS extras seem to work. If Speedscreen on ROM is turned on with _SPEED 1 before loading CP/M-68K the screen output does seem faster. (_SPEED 3 grabs some memory and stops CP/M loading.) If Toolkit II is present on ROM its last line recall key press of ALT-ENTER still works with CP/M-68K. Also the TK2 NFS_USE command allows CP/M-68K to access the devices, e.g. microdrives or ramdisks (if renamed as microdrives by the DEVNAME RAM,MDV QfLash command), on the other QL on my network - this should not be possible! (Can any other CP/M machines use a network or is the QL unique?) Finally, the QDOS CTRL-F5 key press freezes the screen display in CP/M-68K as normal, but generates a block character on screen. All this seems to suggest that Quest may have used some QDOS ROM routines to speed things up. I would like to know whether CP/M-68K will function correctly on a QL converted to QView's 'Minerva' operating system upgrade?

TONY TEBBY - WHERE ARE YOU?

By experimentation I discovered a strange use for the FLP_USE MDV command on my Sandy disk interface. I copied the CP/M-68K boot software onto a 3.5" disk and using microdrive emulation loaded it up. FLP1_ (emulating MDV1_) then became drive A: under CP/M-68K. By using the Quest MDCOPY utility I was able to transfer QDOS files from 3.5" disk to CP/M format 5.25" floppy. I then tried formatting a 720K 3.5" disk with the MDFORMAT command. It produced a 100K CP/M disk, which the operating system thought was a very fast microdrive cartridge, and the remaining 600K or so could still be used for QDOS files! Under QDOS, using the TK2 WSTAT command, I found that CP/M-68K had produced a 100K file called CPM_FILES.

It was possible to copy CPM_FILES to another QDOS disk and then CP/M-68K could still read the new disk as if it had been formatted with MDFORMAT. None of this is documented in Quest's CP/M-68K User Guide or the Sandy disk interface manual so I don't know whether QDOS or CP/M should get the credit. Can Tony Tebby provide any enlightenment?

Michael L. Jackson

ED: Tony Tebby can usually be reached at QJump, 24 King Street, Rampton, Cambs, CB4 4QD. (tel: 0954-50800).

Simon Goodwin, as he writes regularly for QL World, will probably get any post sent there, clearly marked for his attention.

Any more information regarding Quest products and CP/M-68K would be most appreciated.

LETTER

LETTER(S) FROM RICH MELLOR

(I've taken the liberty to string together some comments Rich has made to me in various letters as they may well be of interest to other readers.)

QLTR1 - Very good, especially those articles by a Mr R. Mellor.

Lightning - Special Edition v2.09 works fine with Hack now, even Mortville Manor will run with it switched on, althhough the display is still slightly corrupted for some reason, and the ROM version is a lot faster than the previous ones. When I get time I'll do a follow-up review of Lightning Special Edition.

Machine Code tutorial - I'll try and find time to do Part Two of this, but my studies are taking up a lot of my time at present.

Quest Interfaces - Didn't Tony Tebby do an updated version for this?

Rich

HELPLINE

Like our sibling magazine, QL Adventurers' Forum, we'll be running this as a regular feature. The main difference is that whereas I something about adventures, I know a lot less about the technical side of the QL. So if you see a query in here do write with your answers.

TURBO - WHEN ERROR

This is from Rich Mellor and regards Turbo v2.05, in particular getting the error - trapping to work properly.

"I have tried the following lines in a Turbo program (near the start).

```
10 WHEN_ERROR 1
20  CLS:  INK  7:  PRINT  'AN  ERROR  HAS
OCCURED'\'THE ERROR NUMBER IS';ERNUM%\'THE
LINE NUMBER IS';ERLIN%
30 PAUSE:STOP
40 END_WHEN
```

However when an error does occur in the program, the error is merely reported in the normal way, without the code being called. Is it anything to do with the fact that the above lines occur in the main body of the program, whilst the error is caused by a line within a PROCedure ? I have tried putting the above lines in the offending PROCedure, but to no effect.

MONITORS AND DISSASSEMBLERS

Ian Bruntlett, of QL Forum fame, has written in recommending the dissassembler in the QUANTA library and QMON v2.00 as the best monitor. Any more offers?

QL SCANNERS

Have any of our readers (or people with whom they are in contact) had any experience with Juergen Falkenberg's QL Scanner? The reason I ask is because the members of the Scottish QL Users Group spent a most frustrating day trying to get Alex Beveridge's scanner to work, but with little success. This particular scanner works by having the optical scanner attached to the print head of a printer, which transmits the data back to the QL onto the screen. I have a similar device for the ST, but have never used it. All offers of advice and help to Alex c/o of the Scottish QL User Group.

More next time, Richard

Finance and Business Software Wanted

A PLEA FOR HELP FROM M. L. JACKSON

Though I am a qualified accountant, my interest in computing does not tend towards the latest '386-based IBM compatible personal computers, costing £1000+ and beyond the reach of most small business and home users. Instead the computer which has the greatest appeal for me is the humble QL, still alive and kicking despite Alan Sugar's efforts to give it an honourable burial back in 1986. Available second hand from £30 upwards and new from about £100, depending on additional hardware and accessories supplied with it, the QL is still well worth considering for home users or small businesses concerned to keep track of their finances. Many cheap software packages are still available to encourage purchase of this machine.

In researching this article, I was surprised by the variety of financial and business programs which have been written for the QL. In defining a 'business program' I am excluding a vast range of programs which would be of use to businesses in one way or another, such as Speedscreen or Lightning, Text ^{87} or Editor, Page Designer or Professional Publisher, but are not specifically aimed at businesses only. I have concentrated on accounting packages, stock control, invoice generators, etc., plus home finance programs of interest to individuals, charities or clubs. The range of such programs that has been produced for the QL seems impressive compared to other 'dead' computers (e.g. MSX) and even to more popular 'live' micros (e.g. Amstrad CPC). It must be admitted, however, that only a small number of the programs in the list that follows are being advertised actively by their publishers, which may mean that several will be difficult to obtain.

Many of the programs are unknown to me, other than their small advertisements at the back of Sinclair QL World, which may have appeared only once or twice before the software disappeared without trace. Unfortunately not all QL software gets reviewed, and many would-be purchasers are reluctant to part with their cash until it is confirmed that the software is ready, it works, and it is at least as good as competing products. In the meantime, the publishers are disappointed with the number of orders, lose hope and stop advertising. Consequently the program does not get reviewed. A vicious circle for the small software house.

I would like to write a number of reviews of the programs in the list with a view to publication in QL Technical Review. Obviously, if I were to purchase all of these packages it would cost me several hundred pounds (with no financial reward in return). I need, therefore, the co-operation of the publishers, authors and users of these programs. I would prefer programs to be sent by the publishers for review, with details of cost and availability. Alternatively, users could give software they no longer use or send it on temporary loan. Also I would be prepared to purchase some packages second hand, if CHEAP.

(i) Send the software to me, c/o CGH Services, stating whether you are giving it or loaning it (remember to include your return address if the latter).

(ii) Write to me, c/o CGH Services, if you want to sell the program, stating the price you want.

I would also be interested to hear of any similar products I have left off the list, and readers' comments about the usefulness or otherwise of the programs, or any tips or details of bugs (and solutions?).

Some publishers may be able to supply their programs, but have withdrawn them from general sale. If they are interested in relaunching the products, perhaps after some modification, would they please say so when sending the software for review.

Michael L Jackson

QL BUSINESS AND FINANCE PROGRAMS

1) HOME FINANCE & BUDGETING

New Horizons Software, 'Home Accounts'
Bridgebrook Intek, 'Bank Manager'
Transact (?), 'QL Bank Account'
Data Management, 'QL Finance Manager'
SL Services, 'Home Accounts and Budgets'
Zeataline Projects Ltd., 'Pacioli'
Qmonix (Sharps?), 'Checkbook Manager'
Jay Lewington (Microdrive Exchange), 'Bank'
SD Microsystems, 'QL Bank Account'
PDQL, 'Home Budget'
DJW Software, 'Home Banker V1.2'
DJW Software, 'Home Banker V2.1'
DJW Software, 'Home Banker Plus V3.1'
QL Supersoft, 'Home Budgeting'

NOTE: I already have Sinclair, 'QL Home Finance'.

2) HOME INVESTMENT/ASSET MANAGEMENT

Portfolio Software, 'Stockmarket Manager'
Portfolio Software, 'Insurance Manager'
Maurice Computers, 'QL Stockmarket Investor'
DJW Software, 'Investment Monitor'

3) SMALL BUSINESS ACCOUNTS, STOCK CONTROL, ETC.

Eidersoft, 'Archiver'
Eidersoft/Impaact, 'Integrated Accounts System'
Transform, 'Stock Control'
Sinclair, 'QL Cash Trader'
PDQL, 'Cash Trader'
PDQL, 'PDQ VAT Accounts'
Charringtons, 'QL Sales Order Processing'
SD Microsystems, 'Small Traders Pack'
SD Microsystems, 'Small Traders Pack 2'
PDQL, 'CT Analyser'
PDQL, 'Trading Accounts'
P.C.C., 'Accounter v2.0'
SD Microsystems, 'General Ledger'
SD Microsystems, 'Stock Accounting'
PDQL, 'Private Hire Contracts'
Medic, 'M-Accnts' (if ever published!)

NOTE: I already have Sinclair, 'QL Integrated Accounts' (Sagesoft).

4) SMALL BUSINESS PAYROLL

TR Systems, 'Payroll'
PDQL, 'PDQ Payroll'

5) BUSINESS - MISCELLANEOUS

Software Atelier, 'Project Management Program'
PDQL, 'PDQ Invoice Generator'
SD Microsystems, 'QL Invoicer'
Micr_A_Soft, 'Q_Print (Invoice Printer)'

NOTE: I already have Sinclair, 'Project Planner', 'Decision Maker', and 'Entrepreneur'.

THE SPY EDITOR

Price : £14.95
Supplier: A.R.K. Distribution, Corve Farmhouse, Chale Green, Ventnor, Isle of Wight, PO38 2LA.

The SPY is a multi-file, multi-tasking editor for all QL and CST Thor computers. This means that you can edit several files and run other applications all at once, a very useful feature when developing code.

It must be made clear from the start that the SPY is not a word processor- there is no text justification, no fancy fonts (founts) and other such features found on WPs. It is however possible to produce short memos/letters, but don't expect a replacement for Text87 or DP Editor in terms of printed output. Printer drivers aren't supported, but text can be either 'echoed' or written to the screen (or any other device).

The first thing that struck me about the SPY is just how fast it actually is. Written entirely in assembler and thanks to the use of a lazy screen it is possible to zip through the code at lightning speed (incidentally, it is compatible with Lightning!), and it makes Quill look decidely snail-paced. By using combinations of CTRL and ALT together with the arrow keys, the cursor can be moved along lines in various useful ways, further speeding the task at hand.

The screen is split into three windows - the main editing window which takes up most of the screen, the menu window which occupies just one line at the bottom of the display, and an error window which is hidden behind this menu window, and which overwrites it briefly whenever it is needed. This sensible layout allows for the maximum amount of data to be on screen at any one time.

Commands can either be accessed directly by pressing CTRL and the relevant key, or by pressing F3 which places you in the command menu, as in the PSION suite of programs. These commands are as follows:

R. READ - Reads a new file (of any type) into the SPY.
W. WRITE - Writes the file to any device, including printer.
B. BLOCK - Accesses the block function, of which there can be one in each of the files being edited.
D. DOWN - Downward search for a specified string.
U. UP - Upwards search for a specified string.
K. KERNEL - Diplays useful kernel information on the current file.
S. SET - To change system parameters.
V. VIEW - To look at any other file currently being held in the SPY stack.
P. POKE - extremely useful command to poke a byte at the cursor position. (Full ASCII set from 0 to 255 supported).
M. MOVE - To move quickly to the Top, Bottom, or specified Line of the file.
Z. ZAP - To remove the current file from SPY's stack.
Q. QUIT - Leave SPY.

The five command keys are also used,

F1 - Echoes a file to any device.
F2 - Repeats the previous Search.
F3 - Accesses the Command menu.
F4 - Refreshes the windows (especially useful if you are switching between tasks.
F5 - Insert/Overwrite Mode toggle.
CTRL F1 - Reindex file.

Of particular note is the SEARCH routine which is extremely fast, though the omission of a REPLACE routine can be rather annoying at times (eg when changing all references of MDV to FLP in a long SuperBASIC file).

Any type of file can be entered into SPY, even Quill files which are reputedly uncooperative with some editors (though you lose the layout). Corrupted files can also be read and altered, and this facility has already saved me much heartache, not to mention cash, meaning that the SPY has already payed for itself.

One last point to mention is that two versions are supplied with your disk/micro-cassette, one for monitors, the other for use with televisions. This is an excellent idea, and means you won't have to alter the window defaults yourself (ideal for beginners). The manual explains how to make backups, and is a well written and concise booklet covering all aspects of the program.

Of the four editors I have used on the QL - Quill, Computer One (in their Pascal package), Talent (Assemler Workbench), and SPY - my vote goes unreservedly to ARK's SPY. It is fast, compact, easy to understand, and also very cheap. Definately recommended.

JY Rouffiac.

BITS & PIECES

Machine Code Tutorial correction

Eagle-eyed reader Ray Dawson spotted a small error in Rich Mellor's machine code tutorial. Line 2 of the listing, which currently reads:
MOVEQ #2,D4, should read
MOVEQ #2,D3.
Similarly, the reference at the top of page 22, should be changed from D4 to D3.

And here is the explanation:

On entry to the IO.OPEN routine, D3 should point to the type of open which is made. The specifies whether or not the channel already exists, and whether or not your program will be the only program with access to that file. This is not very important in respect of screen devices and therefore the program works fine despite the fact that D3, not being specifically set, could contain any number when the routine is called!!

Rich would also like to point out that some assemblers have problems with labels which contain a dot, e.g. "SD.SETPA". To overcome this, if your assembler keeps reporting an error, try altering the dot to an underscore, i.e. "SD_SETPA".

ANOTHER PLEA FOR HELP/INFO

Malcolm Smith, in Norway, is interested in expanding his QL system, as he feels that the THOR XVI is unlikely to be supported for much longer.

Among the queries that he raised are:
1) What is the best replacement keyboard, with full-travel, etc.
2) How do you daisy-chain disk drives on the QL. We understand that it can support up to 8 disk drives, but are unsure how they should be connected up.
3) Malcolm also wanted both serial and parallel ports.
4) Finally, he wants to be able to have several EPROM's stuck into his QL, including Toolkit II and Lightning, not to mention the Minerva ROM and extra memory.

ED: A second-hand Sandy Q-Board will provide extra memory, a parallel i/face, Toolkit II and RamDisk. Alternatively the Miracle Trump card has more memory, TKII, Ramdisk, and you can get parallel/serial converters which fit into serl with no hassles. Having TKII on the disk i/face may obviate the need for extra ROM ports, as Lightning could then use the existing ROM port. Minerva fits into existing QL's without too much difficulty. I have no experience with replacement keyboards, but the Schon ones look very nice and are still available. Regarding daisy-chaining disks, any offers of assistance much appreciated!

Minerva Rom - some initial impressions

I purchased my Minerva ROM (v1.64) from QView some weeks ago, when the advert was included in the QUANTA magazine. It sounded just the thing and I sent off for two, one for each of my QL's - an AH and a JS. The package arrived fairly soon after. The two ROM's came in a jiffy bag together with the mdv_ (containing a copy of the ROM image) I had sent off with my order.

I was a bit stumped at the total lack of written instructions, until I had the bright idea of seeing what was on the cartridge QView had sent me. This turned out to be a number of files which could be read as Quill docs. The only drawback to this was that you really need hard copy or two machines, if you wish to refer to the literature whilst changing the ROM's over. As I have two machines, it did save me having to print out the fitting instructions.

The instructions are adequate for more modern machines, but the AH has got me flummoxed somewhat as I have noticed that from one of the original ROM's there is a lead to a resistor some distance away. Perusing the doc's on the cartridge has not produced an answer as to how to deal with that yet. I have written to QView, and also telephoned, but have so far received no answer.

The JS machine was a better prospect, so I converted that one. Locating the correct ROM's is easy as they are all numbered. Great care must be taken when removing the two ROM's but with patience it is not too bad. My real problem was in persuading the new ROM to fit into the left hand ROM slot. It was not easy to persuade the ROM to slide into the base so I had to resort to quite strong arm tactics to get the ROM to fit in. Idid try removing the new ROM and fitting it into the right hand ROM slot. Unfortunately the ROM was in so tight that my courage failed me when the ROM refused to move upwards. So I persisted and eventually managed to get the ROM all the way in into the left hand slot.

On reassembling the machine I turned on and was presented with the Minerva start up screen. This is quite pleasant and gives enough information to enable the user to select the mode in which you start up. The options are the usual TV/Monitor options plus a second screen option in either mode. There is also the facility for starting up as a basic 128K machine if you have extra memory fitted. This is quite a useful feature if you have programs that won't run with extra memory, such as the Psion Chess program.

I have the Taskmaster front-end program which is the basis of my system. It is always present and it allows me to run as many programs as will fit into the memory - currently 798K via the Trump Card from Miracle Systems. Under the old regime, on both AH and JS machines, it was possible to run DP's The Editor from within Taskmaster by loading in the extras required before running the Taskmaster program. The Minerva system will not allow this to happen. It is necessary to run The Editor as a separate background program, much the same as Flashback, selecting it via the CTRL-C key presses. This is O.K. but after a period of time the program seems to interfere with the Quill (which runs O.K. under Taskmaster). Quill also locks up in the Minerva mode all on its own, which is why it has taken me so long to get this to Richard. I had four stabs at getting the thing written, only to find that when nearly finished the beast locked up on me. N.B. Before anyone says that so what, we always get this problem, just remember that the dear old AH machine I have (bought just after they got rid of the rear dongle) behaves impeccably and hardly ever locks up. No heat problems at all and a joy to use. (ED: John's problems don't finish their - I had to copy his file to my printer to get a hard-copy that I then had to re-type. Quill refused to handle the file.)

I have tried various programs to date with the Minerva ROM and find that provided that I do not use the second screen mode, all programs will run under Minerva except for War in the East. Even the much improved version by Rich Mellor will not run. A sad disappointment this, as this program has given me many hours of happy wargaming. A pity it won't run on the Minerva ROM.

On a disappointing note, has anyone managed to contact QView when they have a problem? I have not managed to make contact with them since I bought my ROM despite ringing and writing. I see from the various magazines that they still seem to be in being. If you are out there QVIEW, John Vennard would like to hear from you.

The main difference instantly visible under Minerva is the different colour cursors that appear under the different programs. The SuperBasic cursor is red and flashes at about twice the speed of the AH or JS ROM. Not unpleasant. There is a marked improvement in response times even viewed by the human operator. Loading appears slightly quicker, even using disk drives. Networking takes about the same length of time as under the old ROM's. Noticeably slower than accessing direct.

Under Taskmaster the SuperBasic cursor changes to green and when selecting the file handing facility, the cursor changes to red. Minerva also changes the ink colour to white in Channel 0. Screen colour defaults are as in the old systems.

A number of peculiarities were apparent on first starting up the modified machine. Loading of some programs causes them to either not load at all or give start up screens and then fall over because they meet the problem of a facility not being available. A restart usually sorts this out, but it can be a bit unnerving the first time you experience it. I wouldn't like to buy a new program and use Minerva for the first time with this problem about.

To sum up my experience so far, I have reverted to using the AH machine as the main machine with Taskmaster loaded, and use the second one in the network. This gives the more reliable system as first call on the disk drives without any of the problems produced by the Minerva ROM. A smost of the use for the equipment is semi-business it suits my purposes better this way. Would I purchase a Minerva ROM after my own experience to date? Probably not.

There appears to be no great advantage over the old system as the major new option under the Minerva, the second screen facility, is not usable with the current range of programs available. The extra speed increase available is of no practical advantage to me in my set up.

John Vennard

ED: Well there you have one person's view of their experience with Minerva. I suspect that Stuart McKnight's answerphone is filling up with phone calls, and that as QView is very much a "hobby" rather than high-powered business, the demand for information etc may well have temporarily overwhelmed them.

We'd be very pleased to hear of other people's experiences with Minerva, along with hints and tips on getting the most out of the new operating system. In the meantime, if you want more information contact QView at 29 Carnaby Close, Godmanchester, Cambs, PE18 8EE. (Tel: 0480-412884). The cost of the Minerva ROM is £30 (or £25 to members of QUANTA if they enclose the envelope of their latest issue of QUANTA, complete with address label!). You will also need to enclose on mdv or flp (3.5" or 5.25") a copy of the ROM image of your QL. This can easily be obtained by using the command SBYTES dev$_filename,0,48*1024. I have seen the documentation that comes with Minerva, and the specification is quite impressive.

Among the bug-fixes claimed (i.e. on the hand-out) are the CIRCLE Bug (see circle_bug_bas in our PD Library); much faster graphics - almost as fast as Lightning, faster basic progs; WHEN ERRor and WHEN working; Superbasic TRACE hooks "include single step"; maximum use of integer arithmetic; improved string handling; extended character set and so on. I understand that the changes in the Minerva ROM have caused some TURBO'ed and Q_Liberator'ed compiled programs to crash with early versions of Minerva. For more details of Minerva see Simon Goodwin's article in November QL World.

Richard